1. Podaj definicję kryptosystemu

Kryptosystemem nazywamy piątkę (P, C, K, E, D), gdzie spełnione są następujące warunki:

1. P jest skończonym zbiorem możliwych jednostek tekstu jawnego.
2. C jest skończonym zbiorem możliwych jednostek szyfrogramu.
3. K jest przestrzenią klucza; skończonym zbiorem możliwych kluczy.
4. Dla każdego $k \in K$ istnieje reguła szyfrowania $e_k \in E$ i odpowiadająca jej reguła deszyfrowania $d_k \in E$. Wtedy e_k: P→C i d_k: C→P są funkcjami takimi, że $d_k(e_k(x))=x$ dla każdego $x \in P$. Funkcje e_k, d_k musza być wzajemnie jednoznaczne: $x_1 \neq x_2 \Rightarrow e_k(x_1) \neq d_k(x_2)$, i podobnie dla d_k.

2. Omów pojęcia kryptologia, kryptografia i kryptoanaliza.

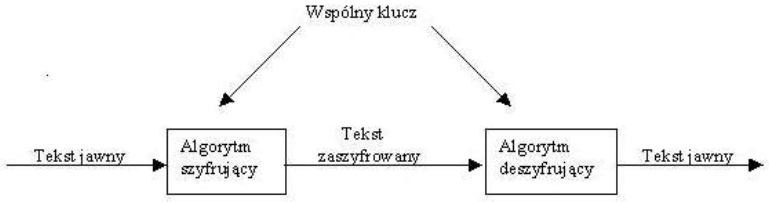
Kryptologia jest to nauka obejmująca kryptografię i kryptoanalizę.

Kryptografia umożliwia dwóm osobom porozumiewanie się przy wykorzystaniu jawnego kanału łączności w taki sposób, że ich przeciwnik nie może zrozumieć treści przekazywanych informacji.

Kryptoanaliza zajmuje się metodami łamania szyfrogramu.

3. Scharakteryzuj kryptosystem symetryczny

Kryptosystem symetryczny polega na tym, że tekst jawny, czyli oryginalny komunikat, ulega przekształceniu, za pomocą algorytmu szyfrującego oraz klucza, na postać zwaną tekstem zaszyfrowanym. Następnie tekst zaszyfrowany jest przekazywany do odbiorcy, i w momencie odbioru może zostać przekształcony z powrotem w tekst jawny za pomocą klucza oraz algorytmu deszyfrującego. Zarówno w przypadku szyfrowania jak i deszyfracji jest używany ten sam klucz.



4. Podaj zasadę działania (przekształcenie szyfrujące) szyfrów: przesuwającego, afinicznego, szyfru Cezara i szyfru willa.

Szyfr przesuwający (shift cipher):

Odwołując się do ogólnej definicji kryptosystemu przyjmujemy $P = C = K = Z_{26}$, gdzie Z_m jest pierścieniem.

Dla klucza $0 \leq k \leq 25$ definiujemy $e_k(x) = (x+k) \bmod 26$, $x \in P$, $d_k(x) = (y-k) \bmod 26$, $y \in C$

W celu szyfrowania tekstu zapisanego z użyciem alfabetu angielskiego (26 liter) numerujemy wpierw kolejne litery przez liczby 0, 1, ..., 25.

Szyfrowanie polega na zastąpieniu danej litery przez literę leżącą *k* pozycji dalej w alfabecie traktowanym jako cykl zamknięty. Deszyfrowanie jest procesem odwrotnym.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Przykład:

Szyfr przesuwający z k=11 i tekst jawny: spotkanie jutro rano

Numerujemy kolejne litery i otrzymujemy ciąg liczb:

18 15 14 19 10 0 13 8 4 9 20 19 17 14 17 0 13 14

Dom każdej liczby dodajemy 11 dokonując ewentualnie redukcji modulo 26 i w wyniku otrzymujemy ciąg liczb:

3 0 25 4 21 11 24 19 15 20 5 4 2 25 2 11 24 25

Zamieniamy teraz liczby na odpowiednie litery otrzymując szyfrogram:

DAZEV LYTPU FECZC LYZ

Przy deszyfrowaniu wykonujemy proces odwrotny: po zamianie liter na liczby odejmujemy 11 modulo 26 i otrzymanym liczbom przyporządkowujemy litery otrzymując w efekcie tekst jawny.

Szyfr przesuwający jest mało bezpieczny, wystarczy sprawdzić wartość klucza k=0-25;

Szyfr afiniczny

Jest on szczególnym przypadkiem szyfru podstawieniowego, który zawiera 26 permutacji z liczby 26 wszystkich permutacji 26 –literowego alfabetu. Funkcja szyfrująca ma postać: $e_k(x) = (ax+b) \bmod 26$, gdzie klucz szyfrujący $k = \{a, b\}$ jest pewną parą liczb z Z_{26}. Dla a = 1 otrzymujemy szyfr przesuwający z parametrem b. Liczba $b \in Z_{26}$ może być w szyfrze afinicznym dowolna, natomiast $a \neq 0$ musi spełniać pewien warunek w celu otrzymania jednoznacznej funkcji deszyfrującej. Jeśli oznaczymy $y=(ax+b) \bmod 26$ to warunkiem jednoznaczności funkcji deszyfrującej jest istnienie jedynego rozwiązania x powyższego równania przy zadanym y. Przekształcenie deszyfrujące ma postać :

$x=d_k(y)=(a^{-1} y-a^{-1} b) \bmod 26$ dla a względnie pierwszych z 26. Klucz deszyfrujący jest jednoznacznie wyznaczony przez k i może być łatwo obliczony stosując algorytm Euklidesa do

obliczania a^{-1} . Jest to kryptosystem symetryczny. Możliwymi wartościami dla a, tzn. takimi, że (a,26)=1 są: a=1,3,5,7,9,11,15,17,19,21,23,25. Szyfr afiniczny ma łącznie 12*26=312 możliwych kluczy.

Przykład:

Szyfr afiniczny z kluczem szyfrującym k={7,3} . Mamy (7,26)=1 i znajdujemy 7^{-1} mod 26 =15. Funkcja szyfrująca ma postać: $e_k(x) = 7x + 3$ natomiast funkcja deszyfrująca $d_k(y) = 15y - 19 = 15y + 7$ gdzie równości rozumiane są modulo 26. Korzystając z tych wzorów szyfrujemy tekst jawny: kryptografia. Po wykonaniu obliczeń otrzymujemy szyfrogram: VSPEGXTSDAHD.

Szyfr Cezara

Szyfr Cezara należy do grupy szyfrów monoalfabetycznych, które polegają na tym, że zamieniają one każdy znak tekstu jawnego na odpowiedni znak kryptogramu, przy czym w całej wiadomości do zamiany każdego znaku jawnego na zaszyfrowany stosuje się odwzorowanie typu *jeden do jednego*.

Szyfr Cezara polega on na przyporządkowaniu każdej literze alfabetu łacińskiego odpowiedniego numeru identyfikacyjnego (np. A=0, B=1 itd.) i dokonaniu przesunięcia numeru każdej litery tekstu jawnego o $k = 3$ pozycje (ma tu miejsce tzw. *przewijanie* - gdy kończy się alfabet przesuwamy się do jego początku). Zakres szyfrowania można oczywiście rozszerzyć na zbiór znaków ASCI (numery od 0-255) lub jakiś inny skończony zbiór *n* znaków. Funkcja szyfrująca będzie się wówczas wyrażała wzorem:

F(a)=(a+k) mod 26 → *F(a)=(a+3)mod26*

Funkcja deszyfrująca wygląda następująco:

F(a)=(a-k) mod 26 → *F(a)=(a-3)mod26*

Przykład:

a	b	c	d	e	f	g	h	i	j	k	l	m
6	7	8	9	10	11	12	13	14	15	16	17	18
G	H	I	J	K	L	M	N	O	P	Q	R	S
n	o	p	q	r	s	t	u	v	w	x	y	z
19	20	21	22	23	24	25	1	2	3	4	5	6
T	U	V	W	X	Y	Z	A	B	C	D	E	F

Tekst jawny	s	z	y	f	r	c	e	z	a	r	a
Szyfr	Y	F	E	L	X	I	K	F	G	X	G

Szyfr Hilla

Szyfr ten został wprowadzony w 1929r. szyfruje on jednocześnie bloki *m* literowe. Niech *m* będzie liczbą naturalną, następnie $P = C = (Z_{26})^m$. Metoda szyfrowania polega na wykorzystaniu m przekształceń linowych *m* liter alfabetu tekstu jawnego; wtedy bloki *m*-literowe traktowane są jako jednostki tekstu jawnego. Dla dowolnego *m* kluczem jest odwracalna mod26 macierz *K* o wymiarach $m \times m$. warunkiem odwracalności tej macierzy jest to, aby jej wyznacznik był liczbą względnie pierwszą z modułem 26. Jeśli $x = (x_1, x_2, ..., x_m)$ i $y = (y_1, y2, ..., y_m)$ są odpowiednio jednostkami tekstu jawnego i szyfrogramu to przekształcenia szyfrujące i deszyfrujące w konwencji mnożenia maja postać: $e_k(x) = y = x K$, $d_k(y) = x = y K^{-1}$.

Zakładając, że d = 2 i $M = m_1m_2$, tekst jawny można zaszyfrować jako $C = E_K (M) = c_1c_2$, przy czym:

$c_1 = (k_{11}m_1 + k_{12}m_2)$ mod n

$c_2 = (k_{21}m_1 + k_{22}m_2)$ mod n

traktując M oraz C jako kolumnowe wektory M = (m1, m2) i C = (c1, c2), możemy zapisać to jako : = EK (M) = KM mod n

przy czym K jest macierzą współczynników:

k_{11}	k_{12}
k_{21}	k_{22}

Deszyfrowania dokonuje się używając macierzy odwrotnej K^{-1}

Przykład 1:

Rozszyfruj kryptogram LJ stosując macierz { 15 20 17 9} → K^{-1}

Macierz 2 x 2 (ciąg)

L-11, I-9 → [15 20 17 9] * [11 9] ➔ $c_1 = (k_{11}m_1 + k_{12}m_2)$ mod 26 ,

$c_2 = (k_{21}m_1 + k_{22}m_2)$ mod 26

➔ (15*11 + 20*9)mod26 =7 = H

➔ (17*11 + 9 *9) mod26 =8 = J

Przykład 2:

LUTY LU TY $y_1=11x_1+23x_2$ $y_2=18x_1+7x_2$

L-11, U-20, T-19, Y-24 (11,20) (19,24)

➔ $[c_1, c_2]$=[11 8 3 7]*[11 20]mod26 → (11*11+8*20)mod26 = 21 = V

(3***11**+**7***20)mod26 = 17 = R

➔ $[c_3, c_4]$=[11 8 3 7]*[19 24]mod26 → (11*19+8*24)mod26 = 11 = L

(3***19**+**7***24)mod26 = 17 = R

K^{-1} =(7 18 23 11) → [7 18 23 11] * [21 17]→ (7*21+18*17)mod26=11=L

→ (23*21+11*17)mod26=20=U

→ [7 18 23 11] * [11 17]→ (7*11+18*17)mod26=19=T

→ (23*11+11*17)mod26=24=Y

5. Omów różnicę i podaj przykłady szyfrów podstawieniowych i przestawieniowych

Szyfry przestawieniowe

Szyfry te zmieniają uporządkowanie bitów lub znaków w danych według pewnego schematu. Zazwyczaj dokonuje się przestawienia za pomocą pewnej figury geometrycznej. Szyfrowanie przebiega więc w dwóch krokach: tekst jawny wpisuje się do figury w sposób określony pewną tzw. *ścieżkę zapisu*, a następnie odczytuje się go według określonego porządku (*ścieżki odczytu*) otrzymując tekst zaszyfrowany. Klucz obejmuje więc figurę geometryczną oraz ścieżki zapisu i odczytu.

Jako pierwszy przykład weźmy prosty **szyfr płotowy**. Litery tekstu jawnego zapisuje się tu tak, aby tworzyły kształt przypominający wierzchołek płotu zbudowanego ze sztachet. Tekst zaszyfrowany otrzymujemy odczytuj¹c kolejne wiersze tak utworzonej konstrukcji. Bardzo często używaną figurą geometryczną jest macierz dwuwymiarowa. Jako przykład szyfru weźmy tzw. **przestawienie kolumnowe**. Tekst jawny zapisuje się do macierzy wierszami. Kryptogram powstaje jako odczyt kolumn w określonym porządku.

Spora część szyfrów przestawieniowych zmienia kolejność znaków tekstu jawnego przy zastosowaniu stałego okresu *t*. Są to tzw. szyfry **okresowo-permutacyjne**. Można je implementować jako przestawienie kolumn macierzy, do której tekst jawny wpisano wierszami. Kryptogram odczytywany jest tu również wierszami.

W szyfrach przestawieniowych (permutacyjnych) elementy tekstu jawnego nie zmieniają się lecz zmienia się ich kolejność w tekście jawnym, co daje w efekcie odpowiedni szyfrogram. Do opisu szyfrów przestawieniowych wygodniej jest użyć liter tekstu jawnego a nie ich odpowiedników liczbowych. Niech $m \geq 2$ będzie ustalona liczbą naturalną, wtedy $P = C = A^m$, gdzie A jest alfabetem tekstu jawnego (np. 26-elementowym zbiorem liter), przestrzeń klucza *K* składa się ze wszystkich permutacji zbioru $\{1, 2, ..., m\}$. Dla ustalonego klucza *k* będącego permutacją π przekształcenie szyfrujące ma postać:

$e_\pi(x_1, x_2, ..., x_m) = (x_{\pi(1)}, x_{\pi(2)}, ..., x_{\pi(m)})$,

a przekształcenie deszyfrujące: $d_\pi(y_1, y_2, ..., y_m) = (y_{\pi}\text{-}1_{(1)}, y_{\pi}\text{-}1_{(2)}, ..., y_{\pi}\text{-}1_{(m)})$, gdzie π^{-1} jest permutacją odwrotną.

Przykład 1:

J O T O

U R J S S B T

T E O A

Szyfr: „JOTOURJSSBTTEOA” . Klucz tego szyfru jest określony wysokości płotu, która w tym przypadku wynosi 3.

Przykład 2:

Dany jest tekst jawny: kryptografia. Słowa te wpisujemy wierszami do macierzy o wymiarach 4 x 3.

1	2	3
K	R	Y
P	T	O
G	R	A
F	I	A

Następnie odczytujemy wpisany tekst kolumnami w kolejności kolumn (2 1 3) otrzymując szyfrogram: „RTRIKPGFYOAA”

Szyfry podstawieniowe

W kryptosystemie opartym na szyfrze podstawieniowym P = C i jest 26-literowym alfabetem. Opis szyfru jest prostszy jeśli wykonujemy bezpośrednie operacje na literach, bez ich kodowania za pomocą liczb. Przestrzeń klucza K składa się ze wszystkich możliwych permutacji zbioru 26-elementowego. Jeśli klucz k = π jest ustaloną permutacją, wtedy przekształcenia szyfrujące i deszyfrujące maja postać: $e_\pi(x) = \pi(x)$, $d_\pi(y) = \pi^{-1}(x)$, gdzie π^{-1} jest permutacją odwrotną.

Przykład:

a	b	c	d	e	f	g	h	i	j	k	l	m
X	N	Y	A	H	P	O	G	Z	Q	W	B	T
n	o	p	q	r	s	t	u	v	w	x	y	z
S	F	L	R	C	V	M	U	E	K	J	D	I

Tekst zaszyfrowany:

MHSVI DPCFO CXTQH VMMCU ASDAF FAYIM XSZX

Tekst odszyfrowany:

Ten szyfrogram jest trudny do odczytania

Przykładami szyfrów podstawieniowych są: szyfr afiniczny, Vigenere’a, szyfr Hilla, szyfr Cezara.

6. Omów różnicę i podaj przykłady szyfrów monoalfabetycznych i polialfabetycznych

Szyfry monoalfabetyczne

Zamieniają one każdy znak tekstu jawnego na odpowiedni znak kryptogramu, przy czym w całej wiadomości do zamiany każdego znaku jawnego na zaszyfrowany stosuje się odwzorowanie typu *jeden do jednego*. Szyfry monoalfabetyczne zamieniają każdy znak uporządkowanego alfabetu jawnego A na odpowiednik znaku uporządkowanego alfabetu szyfru B, zazwyczaj alfabet B powstaje przez prostą zamianę kolejnych znaków alfabetu A. Funkcja f(A)→ B jest wzajemnie jednoznacznym odwzorowaniem każdego znaku alfabetu A na odpowiednik znaku alfabetu B. Kluczem do tego szyfru jest alfabet B lub równoważnie funkcja f. $M=m_1\, m_2$ ➔ $E_k(M)=f(m_1)\, f(m_2)$

Przykładem szyfru są szyfry przesunięcia, podstawieniowe, szyfr Cezara, szyfr afiniczny. Kryptogram w tych szyfrach zachowuje rozkład częstości występowania znaków tekstu jawnego.

Szyfry polialfebetyczne

Stosuje się w nich wiele odwzorowań znaków tekstu jawnego na znaki kryptogramu, przy czym każde odwzorowanie jest z reguły typu *jeden do jednego*, podobnie jak w szyfrach monoalfabetycznych. Szyfry polialfabetyczne ukrywają rozkład częstości przez użycie wielu podstawień. Przykładem szyfru jest szyfr Viegenera, Vernama, Beauforta.

Dla szyfrów tych zachodzą: niech $m \geq 2$ będzie ustaloną liczbą naturalną. Przyjmujemy, że $P = C =K =(Z_{26})^m$. dla klucza $k = \{k_1, k_2, ..., k_m\}$definiujemy przekształcenie deszyfrujące $e_k (x_1, x_2, ..., x_m) = (x_1+k_1, x_2+k_2, ..., x_m+k_m)$ i przekształcenie deszyfrujące $d_k (y_1, y_2, ..., y_m) = (y_1-k_1, y_2-k_2, ..., y_m-k_m)$, gdzie działania arytmetyczne wykonujemy modulo 26. Przy pomocy *m*-literowego klucza k szyfrujemy ciąg *m* liter tekstu jawnego i każda litera może być zaszyfrowana na *m* różnych liter szyfrogramu.

7. Podaj zasadę Kerckhoffa

Zasada Kerckhoffa mówi o tym, że Przeciwnik zna użyty kryptosystem, nie zna natomiast zastosowanych kluczy. Z tego względu kryptosystemy powinny być projektowane tak aby ich bezpieczeństwo opierało się na kluczach a nie na tym, że nie jest znana ich ogólna struktura.

8. Podaj cel i podstawowe typy ataków kryptograficznych

1. Znany tylko tekst zaszyfrowany (ciphertext-only). Celem jest odtworzenie tekstu jawnego lub użytego klucza.
2. Znany jest tekst jawny (known plaintext). Przeciwnik zna pewne pary wiadomości – jawną *x* i odpowiadającą jej zaszyfrowaną wiadomość *y*. Celem jest znalezienie odpowiadającego klucza, który mógłby użyty do deszyfrowania innych wiadomości.
3. Wybrany tekst jawny (chosen plaintext). Przeciwnik ma możliwość szyfrowania wybranych przez siebie wiadomości i uzyskania odpowiadających szyfrogramów. Może się to odbywać w ten sposób, że przeciwnik ma czasowy dostęp do urządzenia szyfrującego lub zleci komuś zaszyfrowanie danej wiadomości. Celem ataku jest uzyskanie klucza szyfrującego.
4. Wybrany tekst zaszyfrowany (chosen ciphertext). Przeciwnik ma okresowy dostęp do urządzenia deszyfrującego. Może wybrać szyfrogram i uzyskać odpowiadający mu tekst jawny. Celem jest uzyskanie klucza szyfrującego.

9. Na czym polega atak brutalny (pełne przeszukiwanie) i od czego zależy jego złożoność

Do problemu przełamania szyfru o n możliwych kluczach mogą być zastosowane dwa podejścia: metoda wyczerpującego przeszukiwania wszystkich kluczy, która pochłania dużo czasu oraz metoda sprawdzania w tablicy wymagająca użycia dużej ilości pamięci. Metoda wyczerpującego przeszukiwania polega na tym, iż dla danego szyfrogramu C i odpowiadającego mu tekstu jawnego M sprawdzane są wszystkie klucze tzn. tekst M szyfrowany jest przy użyciu każdego możliwego klucza w celu otrzymania odpowiadającego mu szyfrogramu C. Zużycie czasu w tej metodzie wynosi 0(n) , natomiast zajętość pamięci 0(1).

10. Podaj i omów metodę kryptoanalizy szyfrów podstawieniowych

Szyfry podstawieniowe: afiniczny, Viegenera, Hilla, Cezara.

Kryptoanaliza szyfru afinicznego:

- podany jest szyfrogram składający się z kilkunastu liter
- otrzymujemy tablicę częstości występowania liter w szyfrogramie
- sprawdzamy jakie litery najczęściej występują w szyfrogramie
- zakładamy pierwsze przypuszczenie , że dana litera odpowiada e i druga litera odpowiada t, gdzie $e_k(x)=ax+b$ jest funkcją szyfrującą o nieznanych parametrach a i b. Rozwiązujemy układ równań. Po rozwiązaniu układu równań sprawdzamy, czy dane przypuszczenie jest spełnione itd.
- Jeżeli przypuszczenie jest spełnione prowadzi do funkcji deszyfującej, która daje tekst jawny

Kryptoanaliza szyfru Hilla:

- zakładamy, że przeciwnik zna wartość m i posiada co najmniej m par jednostek tekstu jawnego i odpowiadających szyfrogramów: $x_j=(x_{1,j} ,..., x_{m,j})$

  $y_j=(y_{1,j} ,..., y_{m,j})$ j=1,...,m

  $y_j=e_k (x_{,j})$ gdzie klucz $K=(k_{i,j})$ jest macierzą o wymiarach m x m. Jeśli utworzymy dwie macierze $X=(x_{,i,j})$ $Y=(y_{,i,j})$ to mamy równanie macierzowe Y=XK. Jeśli macierz X jest odwracalna to przeciwnik może policzyć macierz klucza $K=X^{-1}Y$. Jeśli macierz X nie jest odwracalna należy próbować innych m par tekst jawny i odpowiadający szyfrogram. W przypadku gdy przeciwnik nie zna wartości m i nie jest ona zbyt duża, może wtedy próbować

kolejnych wartości m=2,3..... aż znajdzie klucz. Jeśli m nie jest właściwa to nie będzie właściwej odpowiedniości dla innych par tekst jawny – szyfrogram.

11. Omów szyfr Vigenera i podaj metody kryptoanalizy

Szyfr Vigenere'a poszczególne litery tekstu jawnego mogą być przekształcone na różne litery alfabetu szyfrogramu. Ten kryptosystem należy do kategorii polialfabetycznych. Liczba możliwych kluczy w szyfrze Vigenere'a jest bardzo duża, równa jest 26^m. Dla m.=5 przestrzeń klucza jest większa niż $1{,}1*10^7$. Sprawdzenie takiej ilości kluczy jest zadaniem dla komputera, jednak istnieją metody kryptoanalizy, które umożliwiają złamanie szyfru Vigenere'a w czasie szybszym niż przeszukiwanie całej przestrzeni klucza.

Jeśli w szyfrze Vigenere'a długość użytego klucza równa jest długości tekstu jawnego to nazywamy go *szyfrem z kluczem bieżącym*. Jeśli dodatkowo klucz ten jest losowym ciągiem liter lub bitów oraz klucz jest użyty tylko jeden raz to jest to *szyfr z kluczem jednokrotnym (one time pad).*

Szyfrowanie wiadomości przebiega tu na podstawie dowolnie wybranego słowa kluczowego (hasła). W przypadku znaków ASCI może to być dowolny ich ciąg. Do numeru każdego kolejnego znaku tekstu jawnego dodajemy numer odpowiadającego mu znaku słowa kluczowego i uzyskujemy znak kryptogramu. Gdy słowo kluczowe się skończy, bierzemy je kolejny raz od początku. Dla znaków ASCI szyfr Vigenere'a można przedstawić za pomocą poniższej funkcji:

$F_i(a) = (a+k_i) \bmod 255$

Funkcja deszyfrująca będzie oczywiście wyglądała tak:

$G_i(x) = (x-k_i) \bmod 255$

W szyfrze Viegenera im dłuższe i bardziej skomplikowane jest hasło, tym trudniej odszyfrować tekst utajniony. Z kolei równie łatwo jest zauważyć, że gdy hasło będzie jednoznakowe otrzymamy prosty szyfr monoalfabetyczny. Dla określonego zbioru znaków, będącego dziedziną działań na tekstach możemy stworzyć tzw. tablicę Vigenere'a, która określa nam przesunięcia dla dowolnej kombinacji znaków.

Przykład

Zilustrujmy działanie szyfru na przykładzie zdania "JUTRO JEST SOBOTA". Dla uproszczenia bierzemy pod uwagę tylko litery alfabetu łacińskiego (bez polskich liter). Przyporządkowujemy im kolejno liczby od 0 do 25. Hasłem będzie słowo "KOT". Tekst jawny dzielimy na 3 literowe grupy (pomijając oczywiście spacje) i przesuwamy pierwszą literę każdej grupy o 10 znaków, drugą o 14, a trzecią o 19. Wygląda to mniej więcej tak:

J	**U**	**T**	**R**	**O**	**J**	**E**	**S**	**T**	**S**	**O**	**B**	**O**	**T**	**A**
K	O	T	K	O	T	K	O	T	K	O	T	K	O	T
T	I	M	B	C	C	O	G	M	C	C	U	Y	H	T

Otrzymany kryptogram to następuj¹cy ci¹g liczb: "TIMBCCOGMCCUYHT".

Metody kryptoanalizy szyfru Viegenera:

1. Metoda Kasiskiego
2. Metoda indeksu koincydencji Friedmana

12. Omów krótko jedną z metod kryptoanalizy szyfru Vigenera: metodę Kasiskiego lub metodę indeksu koincydencji Friedmana.

Metoda Kasiskiego

Pierwszym etapem jest określenie długości klucza m. Punktem wyjścia jest obserwacja, że dwa identyczne bloki tekstu jawnego są szyfrowane na te same bloki szyfrogramu, jeśli odległość między początkami tych segmentów jest równa *d*, gdzie *d* jest podzielne przez m. Odwrotnie, jeśli zaobserwujemy dwa identyczne segmenty szyfrogramu (o długości minimum 3 litery – zakładamy, że używamy kluczy o takiej minimalnej długości), wtedy jest duże prawdopodobieństwo, że odpowiadają one identycznym fragmentom tekstu jawnego. Powyższa metoda realizowana jest w następujący sposób: w szyfrogramie szukamy par identycznych fragmentów tekstu i okreslamy odległości między początkami tych segmentów. Jeśli znajdziemy kilka takich par i ich odległości równe są odpowiednio d_1, d_2, ..., wtedy możemy postawić hipotezę, że długość klucza m dzieli największy wspólny dzielnik liczby d_i.

Metoda indeksu koincydencji

Niech $x= x_1, x_2, ..., x_n$ będzie ciągiem n-literowym. Indeksem koincydencji ciągu x, oznaczanym $I_c(x)$ nazywamy prawdopodobieństwo tego, że dwa losowe elementy ciągu x są identyczne. Oznaczmy częstości występowania poszczególnych liter alfabetu w ciągu x przez $f_1, f_2, ..., f_{25}$.

Dwa elementy ciągu x możemy wybrać na $\binom{n}{2}$ sposoby. Dla każdego 0<= i<=25 jest $\binom{f_i}{2}$

sposobów wybrania dwóch elementów równych i. Wtedy oczekujemy, że

$$I_c(x) = \frac{\sum_{i=0}^{25} f_i(f_i-1)}{n(n-1)}$$

13. Podaj warunki na szyfr z kluczem bieżącym i szyfr z kluczem jednokrotnym.

Jeśli w szyfrze Vigenere'a długość użytego klucza równa jest długości tekstu jawnego to nazywamy go *szyfrem z kluczem bieżącym*. Jeśli dodatkowo klucz ten jest losowym ciągiem liter

lub bitów oraz klucz jest użyty tylko jeden raz to jest to *szyfr z kluczem jednokrotnym (one time pad).*

14. Podaj zasadę szyfrowania strumieniowego. Co jest głównym elementem

Szyfry strumieniowe: dzielą tekst jawny na znaki lub bity, a następnie każdy i-ty element tekstu jawnego jest szyfrowany kluczem z_i należącym do strumienia kluczy. Taki ciąg klucza jest generowany przez tzw. ziarno, by zaszyfrować tekst jawny musimy mieć strumień tej samej długości, co tekst jawny. Do wygenerowania ciągu strumienia klucza używa się parametru – ziarna, które stanowi klucz szyfru strumieniowego. Jest on generowany przez funkcję (Z_i), którą nazywamy generatorem ciągu klucza.

ciąg klucza $z=z_1z_2...$,

tekst jawny $x=x_1x_2...$

szyfrogram $y=y_1y_2...=e_{z1}(x_1)e_{z2}(x_2)...$

ciąg tekstu jawnego: $z_i=f_i$ (k, x_1, ..., x_{i-1}).

15. Omów pojęcia szyfr strumieniowy okresowy i synchroniczny

Szyfr strumieniowy przetwarza elementy wejściowe w sposób ciągły, produkując jednocześnie kod wyjściowy. Tekst dzielony jest na znaki lub bity a następnie każdy kolejny element jest szyfrowany kluczem należącym do strumienia kluczy (każdy element jest szyfrowany innym kluczem).

Szyfr **strumieniowy** jest okresowy, jeżeli strumień klucza powtarza się po d znakach, dla danego ustalonego d. w przeciwnym razie szyfr jest nieokresowy. Do okresowych szyfrów okresowych należą szyfry generowane przez maszyny rotorowe i Hagelina. Natomiast szyfr Vernama i szyfry z bieżącymi kluczami są nieokresowymi szyframi strumieniowymi.

Okresowy szyfr o krótkim okresie można traktować jako szyfr strumieniowy, ponieważ znaki tekstu s± szyfrowane jeden za drugim, a sąsiednie znaki przez inną część klucza. Gdy okresy są krótkie, wówczas szyfr jest bardziej podobny do słabego szyfru blokowego niż do szyfru strumieniowego, a jego słabość wynika stąd, że zaszyfrowane znaki nie wpływają na szyfrowanie wszystkich znaków w bloku. Natomiast gdy długość okresu wzrasta, wówczas szyfr staje się bardziej podobny do szyfru strumieniowego.

Istnieją dwie różne metody realizacji szyfrowania strumieniowego.

Szyfry **strumieniowe synchroniczne**

W szyfrach tego typu strumień klucza jest generowany niezależnie od strumienia wiadomości. Oznacza to, że jeśli znak szyfrowanego tekstu zostanie zgubiony podczas transmisji, to nadawca i odbiorca muszą ponownie zsynchronizować swoje generatory kluczy przed wznowieniem transmisji. Proces ten powinien być zrealizowany w sposób zapewniający brak powtarzalności. Przykłady szyfrów strumieniowych synchronizujących się: szyfr Vigenere'a z okresem d , maszyna obrotowa z t mechanizmami obrotowymi , maszyna Hagelina z t kołami, każde po pi kołków, szyfr z bieżącym kluczem , szyfr Vernama

16. Omów pojęcia ziarno i okres generatora klucza

Ciąg klucza (z_i) generowany jest przez *ziarno* k, które można uważać jako „klucz" szyfru strumieniowego oraz przez określone funkcje f_i, które mogą zależeć od wcześniejszego ciągu tekstu jawnego: $z_i=f_i$ (k, x_1, ..., x_{i-1}).

Rodzinę funkcji (f_i) nazywamy *generatorem ciągu klucza.* Ciąg klucza (z_i) szyfruje teraz ciąg wiadomości jawnych (x_i) według określonych funkcji szyfrujących dając $y_i=ez_i(x_i)$. W procesie szyfrowania obliczamy zatem kolejno z_1, y_1, z_2, y_2, ...W procesie deszyfrowania obliczamy kolejno

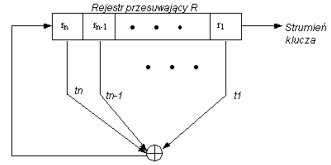
z_1,x_1,z_2,x_2, ...Szyfry blokowe można uważać jako szczególny przypadek szyfrów strumieniowych, gdzie ciąg klucza jest stały $k=z_i$ dla wszystkich i=1,2,... .

17. Narysuj schemat liniowego rejestru przesuwającego ze sprzężeniem zwrotnym

Liniowe rejestry przesuwające ze sprzężeniem zwrotnym (LFSR) – rejestr ten o *n* stanach składa się z rejestru przesuwającego R=(r_n, r_{n-1}, ..., r_1) i rejestru sprzężeń T=(t_n, t_{n-1}, ..., t_1). Każdy element reprezentuje jedną cyfrę dwójkową. W każdym kroku bit r_1 jest przyłączany do łańcucha klucza, bity r_n, ..., r_2 są przesuwane w prawo a nowy bit obliczony z wartości T i R jest wprowadzany z lewej strony rejestru. Następny stan rejestru R'=(r'$_n$, r'$_{n-1}$, ..., r'$_1$) jest obliczony ze wzoru:

Rejestr LFSR może wygenerować pseudolosowy ciąg bitów 2^n-1. By to osiągnąć muszą być wykonane wszystkie 2^n-1 cykle. Warunek taki będzie spełniony, jeżeli wielomian

$$T(x) = t_n x^n + t_{n-1} x^{n-1} + ... + t_1 x + 1$$



utworzony z elementów wartości T zwiększonych o 1 jest wielomianem pierwotnym. Wielomianem pierwotnym stopnia n-tego nazywamy nieredukowalny wielomian, który dzieli wielomian x^{2n-1}+1, ale nie wielomiany x^d+1 dla każdego d dzielącego 2^{n-1}. Szczególnie przydatne są trójmiany pierwotne o postaci $T(x)=x^n+x^\alpha+1$. Strumień bitów wiadomości $M=m_1m_2...$ jest szyfrowany przez obliczanie , przy czym k_i są kolejnymi elementami strumienia

klucza, który został wygenerowany. Deszyfrowanie takiego klucza odbywa się dokładnie tak samo. Sekwencje sprzężeń T można łatwo określić znając wiadomość niezaszyfrowaną i jej zaszyfrowany odpowiednik.

Słabość metody LFSR jest spowodowana liniowością równania. Lepsze wyniki daje zastosowanie przekształceń nieliniowych. Za przykład może służyć metoda OFM – zwana trybem sprzężenia zwrotnego bloków wyjściowych. Rejestr LFSR służy jako wejście do algorytmu E_B szyfru blokowego z kluczem B. Podczas i-tej iteracji jest realizowany algorytm

$$k_i' = k_{i-1}' + 1$$

$E_B(R)$, młodszy znak bloku wyjściowego jest wygenerowanym znakiem klucza K_i. Otrzymany w wyniku algorytmu blok znaków jest ponownie umieszczany w rejestrze R, przy czym każdy element k_i jest znakiem a nie bitem. Wiadomość wejściowa jest dzielona na strumień znaków i szyfrowana z jednoczesnym generowaniem klucza.

18. Scharakteryzuj szyfry strumieniowe samosynchronizujące się

Szyfry strumieniowe samosynchronizujące się

Każdy znak klucza jest tu generowany na podstawie stałej liczby n wcześniej zaszyfrowanych znaków. Jeśli zatem podczas transmisji jakiś zaszyfrowany znak zostanie zgubiony lub

$$r'_n = TR = \sum_{i=1}^{n} t_i r_i \bmod 2 = t_1 r_1 \oplus t_2 r_2 \oplus \ldots \oplus t_n r_n$$

wstawiony, to powstały błąd podlega propagacji przez n następnych znaków. Jednakże po odebraniu n poprawnie zaszyfrowanych znaków synchronizacja zostanie ponownie osiągnięta. Przykładowe szyfry samosynchronizujące się: szyfr z kluczem samogenerującym (autokey cipher) , metoda sprzężenia zwrotnego

19. Podaj ideę i przykłady szyfrów blokowych

Szyfr blokowy przetwarza jeden blok tekstu wejściowego za jednym razem, produkując jeden blok wyjściowy na każdy blok wejściowy. Każdy z tych bloków szyfrowany jest tym samym kluczem i zwykle składa się z kilkunastu znaków, choć nie jest to regułą. Do takich szyfrów zaliczamy szyfry podstawieniowe proste lub homofoniczne, mimo iż jednostką szyfrowania jest jeden znak. Cechą która pozwala sklasyfikować je jako szyfry blokowe jest fakt iż dla każdego znaku stosuje się ten sam klucz. Przykłady szyfrów blokowych wraz z rozmiarami bloków:

Szyfr przestawieniowy z okresem d (transpozycja)	d znaków
Szyfr z prostym zastępowaniem	1 znak
Szyfr z homofonicznym zastępowaniem	1 znak
Szyfr Playfaira	2 znaki
Szyfr Hilla dla macierzy d x d	d znaków
Szyfr DES	64 bity
Szyfr ekspotencjalny mod n	$\log_2$ n bitów (zalecz się 664 bity)
Szyfr plecakowy o długości n	n bitów (zalecz się 200 bitów)

Wybrane cechy szyfrów blokowych

- Stosując szyfrowanie blokowe w sposób bezpośredni uzyskuje się trochę większą szybkość niż w metodach strumieniowych. Jest tak dlatego, ponieważ jest wymagane tylko 1-krotne wykonanie algorytmu szyfrowania dla n znaków.
- Błędy transmisji występujące w jednym bloku nie mają wpływu na inne bloki.
- Szyfrowanie blokowe może być bardziej wrażliwe na kryptoanalizę niż szyfrowanie strumieniowe. Ponieważ identycznym blokom tekstu szyfrowanego odpowiadają identyczne bloki tekstu zaszyfrowanego, zatem np. bloki pustych znaków lub słów kluczowych mogą być łatwo zidentyfikowane przez kryptoanalityka przy ataku z tekstem jawnym.

Szyfry blokowe można uważać jako szczególny przypadek szyfrów strumieniowych, gdzie ciąg klucza jest stały $k=z_i$ dla wszystkich i=1,2,... .

20. Podaj podstawowe tryby pracy szyfrów blokowych

Istnieją cztery podstawowe tryby pracy szyfrów blokowych:

Elektroniczna książka kodowa (ECP) – W szyfrach tych błędy transmisji występujące w jednym bloku nie mają wpływu na inne bloki. Ponieważ identycznym blokom szyfrowanego tekstu odpowiadają identyczne bloki tekstu zaszyfrowanego, bloki pustych znaków lub słów kluczowych mogą być łatwo zidentyfikowane przez kryptoanalityka przy ataku z tekstem jawnym. Przy szyfrowaniu blokowym krótkie bloki muszą być uzupełniane pustymi znakami lub zerami, co ułatwia pracę kryptoanalitykom. W systemach baz danych szyfrowanie każdego pola lub rekordu jest niedopuszczalne. Uzupełnianie krótkich pól pustymi znakami lub zerami zwiększa zapotrzebowanie na rozmiar pamięci, co zwiększa możliwość złamania szyfru. W szyfrowaniu tą metoda istnieje większe prawdopodobieństwo występowania powtórzeń. Gdy każdy blok jest niezależnie szyfrowany tym samym kluczem, wówczas istnieje możliwość zastąpienia jednego bloku innym.

Wady: wrażliwe na kryptoanalizę, brak odporności na wstawianie lub kasowanie bloków – zmiany te nie mają wpływu na sąsiednie bloki.

Wiązanie bloków zaszyfrowanych – szyfry blokowe mogą być bardziej odporne na kryptoanalizę i podmianę tekstów po zastosowaniu tzw. wiązania bloków. Przed zaszyfrowaniem każdego bloku tekstu źródłowego M_i, do tego bloku na wolne, nie wykorzystywane pozycje bitowe umieszcza się kilka bitów z poprzednio zaszyfrowanego już bloku C_i. Technika ta chroni przed wstawianiem, kasowaniem i modyfikacją tekstu źródłowego.

Metoda wiązania bloków zaszyfrowanych (cipher block chaining – CBC) – do rejestru przesuwającego ładowany jest przez pętlę sprzężenia cały zaszyfrowany blok, którego znaki wraz ze znakami następnego zaszyfrowanego bloku uczestniczą w obliczaniu różnicy symetrycznej. Następnie wynik różnicy symetrycznej jest przesyłany do szyfru blokowego realizującego algorytm E_K z kluczem K

$$C_i = E_K(M_i \oplus C_{i-1})$$

przy czym $C_0=I_0$. Deszyfrowanie wykonuje się obliczając

$$D_K(C_i) \oplus C_{i-1} = D_K(E_K(M_i \oplus C_{i-1})) \oplus C_{i-1} = (M_i \oplus C_{i-1}) \oplus C_{i-1} = M_i$$

W związku z powyższym algorytmem szyfrowana błędy, które mogą wystąpić podczas transmisji mają wpływ najwyżej na dwa bloki. Jednocześnie każdy zaszyfrowany blok funkcjonalnie zależy od wszystkich poprzednio zaszyfrowanych , więc cechy statystyczne tekstu źródłowego są rozprzestrzenione w całym szyfrogramie, co utrudnia kryptoanalizę.

Zalety: Odporna na wszelkie rodzaje działań destruktywnych, jest skuteczna – do zaszyfrowania jednego bloku tekstu źródłowego jest potrzebne tylko jednokrotne wykonanie algorytmu szyfrowania blokowego E_K, chroni przed analizą czasowo-pamięciowa.

Wady: problem z ochroną krótkich rekordów, ponieważ nie ma sprzężenia zwrotnego z poprzednim rekordem.

Sprzężenie zwrotnego wyjścia AFB – na podstawie klucza i wartości początkowej jest generowany pseudolosowy ciąg bitowy. Generujemy ciąg bloków do podklucza i podobnie jak w szyfrze strumieniowym szyfrowanie polega na dodaniu do tekstu jawnego strumienia klucza V. Tak samo postępujemy przy deszyfrowaniu.

Zalety: jeszcze mniejsza propagacja błędów, ciąg klucza można otrzymać przed wygenerowaniem tekstu jawnego – deszyfrowanie jest wtedy szybsze.

Sprzężenie zwrotne szyfrogramu (CLP) – kolejny blok szyfrogramu otrzymujemy przez dodanie mod2 przeszyfrowanego poprzedniego szyfrogramu z aktualnym blokiem tekstu jawnego.

Istnieją jeszcze dodatkowe tryby pracy szyfrów blokowych:

- Wiązanie bloków tekstu jawnego
- Sprzężenie zwrotne tekstu tajnego
- Wiązanie bloków zaszyfrowanych różnic tekstu jawnego
- Sprzężenie zwrotne wyjściowe z funkcją nieliniową

21. Omów algorytm blokowy DES. Podaj długość bloku, długość klucza, ilość rund, podstawowe operacje

Standard Szyfrowania Danych DES jest jednym z bardziej skomplikowanych algorytmów szyfrowania. Jest to symetryczny algorytm szyfrujący (ten sam klucz jest używany do szyfrowania i deszyfrowania), zaliczany do szyfrów kaskadowych,. Został on celowo opublikowany w 1977 roku przez Narodowe Biuro Normalizacji w USA. Chodziło o przekonanie potencjalnych użytkowników, że bezpieczeństwo metody nie tkwi w tajności tej budowy, ale w konstrukcji odpornej na kryptoanalizę. Mimo tego, iż DES istnieje już ponad 20 lat, wciąż uważany jest za jeden z najbardziej nowoczesnych szyfrów, a ponadto jest najszerzej stosowany w prawie wszystkich rodzajach łączności cyfrowej i przechowywania danych.

Zasada działania DES

DES jest czasem nazywany iterowanym szyfrem blokowym. Oznacza to, że dane są tu szyfrowane w porcjach o określonej wielkości bloku. DES ma blok 8-bajtowy, czyli algorytm pobiera 8 bajtów tekstu i wytwarza 8 bajtów szyfrogramu. Jeżeli wiadomość jest dłuższa niż 8 bajtów, a większość wiadomości jest właśnie taka, to algorytm szyfruje 8 bajtów naraz. Z drugiej strony łącza algorytm deszyfruje szyfrogram w porcjach 8-bajtowych. Iterowany oznacza, że algorytm składa się z powtarzających się prostych funkcji. DES ma 16 iteracji. Większa liczba iteracji lub etapów zapewnia większe bezpieczeństwo. Jednakże DES jest tak zbudowany, że 16 iteracji wystarcza, aby algorytm mógł być złamany tylko przez łamanie brutalne, dodanie dalszych iteracji nie poprawia zbytnio bezpieczeństwa dawanego przez algorytm.

Długość klucza w algorytmie DES jest w niektórych przypadkach niewystarczająca. Dlatego też poddano go modyfikacjom, tak aby zwiększyć ilość możliwych kluczy. Takim zmodyfikowanym DES-em jest DESX. Użyto w nim prostej metody, która utrudnia atak poprzez poprzez systematyczne przeszukiwanie, zwanej metodą Whiteninga. Dla zaszyfrowania bloków składających się z 64 bitów używa się trzech kluczy.

Funkcji szyfrującej składającej się z 16 iteracji zwanych rundami lub cyklami. W każdym cyklu stosuje się kombinację podstawiania i permutacji z udziałem podkluczy

Algorytm wykorzystuje operacje logiczne XOR (przesunięcia logiczne) na liczbach 64 bitowych. DES szyfruje bloki złożone z 64 bitów, odpowiada to 8 literom ASCII, każda zaopatrzona w bit parzystości. Klucze składają się również z 64 bitów, przy czym 8 bitów jest bitami parzystości. W rzeczywistości można określić jedynie 56 bitów, reszta jest generowana automatycznie. W każdej rundzie dokonywane są te same obliczenia, na wynikach obliczeń z

poprzedniej rundy i specjalnym podkluczu generowanym z 64 bitowego klucza. Przed pierwsza runda i po ostatniej bity są permutowane w ustalony sposób. Dla uzyskania podkluczy usuwamy najpierw 8 bitów parzystości zawartych w kluczu. Z pozostałych 56 bitów tworzonych jest 16 podkluczy, każdy składający się z 48 bitów.

22. Podaj różnice między algorytmami blokowymi DES i IDEA.

IDEA

Jest to międzynarodowy algorytm szyfrowania danych. Został wynaleziony w 1991 roku przez Jamesa Massey□a i Xuejia Lai□a w ETH w Zurychu. Ma podobną ogólną strukturę jak algorytm DES. Jest iterowanym szyfrem blokowym o 64-bitowym rozmiarze bloku i 128-bitowym kluczu. Ma jedynie 8 iteracji w porównaniu z 16 iteracjami algorytmu DES, ale każda iteracja algorytmu IDEA działa tak, jak by była iteracją podwójnego DES. Dla większości mikroprocesorów programowa implementacja algorytmu IDEA jest szybsza od oprogramowania będącego implementacją algorytmu DES.

W algorytmie IDEA klucz jest ponad dwukrotnie dłuższy niż klucz algorytmu DES; klucz ten jest nawet dłuższy od klucza używanego w potrójnym DES. Algorytm IDEA jest znacznie szybszy od potrójnego DES. Z powodu młodego wieku, nie można powiedzieć iż IDEA jest algorytmem super-bezpiecznym. Nad jego bezpieczeństwem i nad ewentualnymi sposobami jego złamania trwają dopiero prace badawcze.

DES

Standard Szyfrowania Danych DES jest jednym z bardziej skomplikowanych algorytmów szyfrowania. Jest to symetryczny algorytm szyfrujący (ten sam klucz jest używany do szyfrowania i deszyfrowania), zaliczany do szyfrów kaskadowych,. Został on celowo opublikowany w 1977 roku przez Narodowe Biuro Normalizacji w USA. Chodziło o przekonanie potencjalnych użytkowników, że bezpieczeństwo metody nie tkwi w tajności tej budowy, ale w konstrukcji odpornej na kryptoanalizę. Mimo tego, iż DES istnieje już ponad 20 lat, wciąż uważany jest za jeden z najbardziej nowoczesnych szyfrów, a ponadto jest najszerzej stosowany w prawie wszystkich rodzajach łączności cyfrowej i przechowywania danych. DES jest czasem nazywany iterowanym szyfrem blokowym. Oznacza to, że dane są tu szyfrowane w porcjach o określonej wielkości bloku. DES ma blok 8-bajtowy, czyli algorytm pobiera 8 bajtów tekstu i wytwarza 8 bajtów szyfrogramu. Jeżeli wiadomość jest dłuższa niż 8 bajtów, a większość wiadomości jest właśnie taka, to algorytm szyfruje 8 bajtów naraz. Z drugiej strony łącza algorytm deszyfruje szyfrogram w porcjach 8-bajtowych. Iterowany oznacza, że algorytm składa się z powtarzających się prostych funkcji. DES ma 16 iteracji. Większa liczba iteracji lub etapów zapewnia większe bezpieczeństwo. Jednakże DES jest tak zbudowany, że 16 iteracji wystarcza, aby algorytm mógł być złamany tylko przez łamanie brutalne, dodanie dalszych iteracji nie poprawia zbytnio bezpieczeństwa dawanego przez algorytm.

Długość klucza w algorytmie DES jest w niektórych przypadkach niewystarczająca. Dlatego też poddano go modyfikacjom, tak aby zwiększyć ilość możliwych kluczy. Takim zmodyfikowanym DES-em jest DESX. Użyto w nim prostej metody, która utrudnia atak poprzez poprzez systematyczne przeszukiwanie, zwanej metodą Whiteninga. Dla zaszyfrowania bloków składających się z 64 bitów używa się trzech kluczy.

23. Co oznacza potrójny DES

W potrójnym DES-ie używa się dwóch kluczy. Tekst jawny szyfrowany jest najpierw kluczem pierwszym, następnie wynik jest deszyfrowany kluczem drugim a na koniec następuje ponowne zaszyfrowanie kluczem pierwszym.

24. Podaj znane ataki kryptograficzne na szyfry blokowe

Kryptoanaliza różnicowa 2^{47} wybranych bloków tekstu jawnego: w metodzie tej sprawdzane są te pary szyfrogramów, których odpowiednie teksty jawne charakteryzują się pewnymi szczególnymi różnicami. Te szczególne różnice w parach tekstów jawnych pozwalają z dużym prawdopodobieństwem określić wartości różnic w pewnej liczbie iteracji. Rozkład takich różnic wraz z prawdopodobieństwem jego wystąpienia nazywamy charakterystyką różnicową. Charakterystyki pozwalają generować zbiór prawdopodobnych kluczy, z którego wybierany jest klucz właściwy.

Kryptoanaliza liniowa 2^{43} tekstu jawnego

Pełne przeszukiwanie 2^{56} kluczy w 3 dni : metoda wyczerpującego przeszukiwania polega na tym, iż dla danego szyfrogramu C i odpowiadającego mu tekstu jawnego M sprawdzane są wszystkie klucze tzn. tekst M szyfrowany jest przy użyciu każdego możliwego klucza w celu otrzymania odpowiadającego mu szyfrogramu C. Zużycie czasu w tej metodzie wynosi 0(n) , natomiast zajętość pamięci 0(1).

25. Podaj założenia i cele konkursu AES

Stworzenie algorytmu mającego zastąpić DES. wg założeń AES ma być:

- Bardzo silnym, symetrycznym szyfrem blokowym do nie klasyfikowanych zastosowań rządowych i komercyjnych
- Bardziej efektywny i bardziej bezpieczny niż potrójny DES
- Długość klucza 128, 192, 256 bitów

- Długość bloku 128(64, 80, 128 lub inna opcjonalnie)
- Publicznie przedstawiony i oceniony
- Nie wymagający opłacania praw autorskich

Zgłoszone algorytmy będą oceniane w kategoriach:

- Bezpieczeństwo
- Koszty
- Algorytm i implementacja
- Wymagania licencyjne

AES ma być wybrany w całkowicie otwartym procesie tzn. każdy miał prawo zgłosić swojego kandydata, wszystkie algorytmy mogą być publicznie komentowane i oceniane. Ostatecznie wybrany algorytm zostanie przedstawiony ministrowi handlu.

26. Scharakteryzuj kryptosystem asymetryczny

Podstawą algorytmu asymetrycznego jest jawność klucza. Używa się dwóch lub więcej kluczy i występują one w parach. Jeden klucz używany jest do szyfrowania a drugi do deszyfrowania. Opublikowanie jednego z kluczy występującego w parze praktycznie nie zdradza drugiego z klucz, nawet gdy można w tym celu wykonać dość złożone obliczenia. Zwykle jeden z kluczy w parze jest powszechnie dostępny i może to być klucz szyfrujący lub deszyfrujący, w zależności od zamierzonych zastosowań i jest on nazywany kluczem publicznym. Drugi klucz jest trzymany w tajemnicy przez jego posiadacza i jest nazywany kluczem prywatnym.
Algorytmy asymetryczne to między innymi: szyfry wykładnicze (Pohliga-Hellmana, RSA, szyfry plecakowe (Marklego-Hellmana, Grahama-Shamira, szyfr Shamira do kontroli autentyczności), algorytm ElGamala.

27. Podaj przykłady kryptosystemów klucza publicznego. Na jakich problemach bazuję się te kryptosystemy

Algorytmy oparte o klucz publiczny powstały w celu stworzenia możliwości szyfrowania, dzięki której informacje zaszyfrowane za pomocą jednego klucza mogą być odszyfrowane za pomocą drugiego niezależnego klucza. Istnieją wiele systemów opartych o klucz publiczny do których należą m.in.: RSA, DSS, ElGamal, Diffiego-Hellmana. Algorytmy te opierają się na teorii liczb. Algorytmy oparte o klucz publiczny są łatwe do złamania, gdyż napastnik dysponuje kopią klucza publicznego użytego do zaszyfrowania wiadomości. Dodatkowym ułatwieniem dla atakującego jest fakt, iż wiadomość zawiera zapewne informacje o algorytmie użytym do jej zaszyfrowania. Ataki na algorytmy oparte o klucz publiczny podzielić można na dwie grupy: ataki wykorzystujące rozkład na czynniki pierwsze oraz ataki algorytmiczne.

Ataki oparte na rozkładzie na czynniki pierwsze są najpopularniejszymi atakami na algorytmach wykorzystującymi klucz publiczny ponieważ iż opierają się naprzystępnej teorii. Atak tego typu polega na próbie obliczenia klucza prywatnego na podstawie odpowiadającego mu klucza publicznego. Druga klasa ataków na systemy kryptograficzne bazujące na kluczu publicznym opiera się na odnajdywaniu błędów oraz słabych punktów teorii na której oparte są te systemy.

28. Przedstaw algorytm Diffiego-Hellmana. Do czego służy ten algorytm

Algorytm Diffie'go Hellman'a opiera się na problemu logarytmu dyskretnego. Mamy pewne ciało N elementowe o kluczu p, gdzie p jest liczba pierwszą. Liczymy potęgę, ale znalezienie wykładnika jeśli znamy podstawę i potęgę jest problemem skomplikowanym.
Algorytm ten służy do dystrybucji kluczy a nie do szyfrowania. Nie trzeba komunikować się tajnie by przekazać klucz, za pomocą tego algorytmu możemy ten klucz ustalić jawnie.
Działanie algorytmu:
1. Obydwie strony uzgadniają w sposób jawny dwie liczby, odpowiednio duże i całkowite n i g, $g \in (1,n)$;
By algorytm był bezpieczny musi spełniać dwa warunki:
- n i $(n\text{-}1)/2$ muszą być pierwsze, n powinna być co najmniej 512-bitowa,
- g musi być pierwiastkiem pierwotnym modulo n, jest generatorem ciała modulo n, tzn., że wszystkie potęgi $g(1,n)$, tworzą wszystkie elementy tego ciała.

3. A wybiera odpowiednio dużą liczbę x, zna ją tylko A i oblicza
$g^x \bmod n = X$
3. B postępuje podobnie – wybiera duże y i oblicza $g^y \bmod n = Y$
4. Obydwie strony wymieniają się tymi informacjami, które obliczyli;
5. A następnie oblicza $Y^x \bmod n$, B zaś $X^y \bmod n$ – w rezultacie otrzymują te same liczby – jest to klucz.

Osoba śledząca zna n, g i Y, nie zna x i y. By wyznaczyć klucz musi podać x i y, czyli wyznaczyć logx lub logy, co jest nie możliwe w sensownym czasie, przy tak dużych liczbach. Algorytm Diffie'go i Hellman'a można stosować dla dowolnej liczy osób. Zwiększa się wtedy liczba kroków postępowania.

29. Przedstaw algorytm RSA. Które parametry stanowią klucz prywatny, a które publiczny

RSA jest systemem bazującym na kluczu publicznym. System ten nadaje się zarówno do szyfrowania informacji, jak i do tworzenia systemów podpisów cyfrowych. Podpisy cyfrowe

mogą być używane do potwierdzania autentyczności informacji cyfrowych. Klucze w systemie RSA mogą mieć dowolną długość, zależną od konkretnej implementacji systemu.

Wybór kluczy:

- losowo wybieramy dwie duże liczby pierwsze *p, q*;
- losowo dobieramy liczbę *e* tak, aby *e* i *(p-1)(q-1)* były względnie pierwsze, (wybieramy losowo *e* i przy użyciu algorytmu Euklidesa obliczamy NWD(*e,(p-1(q-1)*);
- za pomocą algorytmy Euklidesa znajdujemy d, takie że: *e*d=1 mod (p-1)(q-1)*
- obliczamy *n:=p*q* i kasujemy liczby *p,q* tak, aby nie pozostał po nich żaden ślad

[e,n] jest wygenerowanym kluczem publicznym, *[d,n]* jest kluczem prywatnym.

Szyfrowanie: szyfrowane mogą być liczby m<n:

$E_{[e,n]}(m)=m^e \bmod n$.

Szyfrowany tekst, będący ciągiem bitów, jest traktowany jako zapis binarny liczby m. Tak więc tekst jawny nie może być zbyt długi, bo liczba m nusi być mniejsza niż n.

Deszyfrowanie: $D_{[d,n]}(c)=c^d \bmod n$

30. Na czym opiera się bezpieczeństwo RSA? Czego poszukuje kryptoanalityk? Jakie wielkości parametru n uznaje się za bezpieczne?

Bezpieczeństwo RSA opiera się na tym, że aby złamać szyfr należałoby rozłożyć n na czynniki pierwsze. Problem rozkładu liczby na czynniki pierwsze jest problemem klasy NP., ale nie wiadomo, czy jest to problem NP-zupełny, co świadczyłoby o jego trudności. Z drugiej jednak strony najlepsze obecnie znane algorytmy rozkładające liczby na czynniki pierwsze wymagają dość dużych czasów obliczeń. Powoduje to, że tylko dla stosunkowo niewielkich wartości n metoda ta może być stosowana. Głównym elementem zapewniającym mu bezpieczeństwo jest problem faktoryzacji dużych liczb. Zakładając, że dysponujemy siecią miliona komputerów z których każdy wykonuje milion kroków na sekundę, to faktoryzacja liczby n o długości 1024 bitów zajęła by 1010 lat [3]. Nie wiadomo jednak jak postępować będzie rozwój technik faktoryzacji i na jak długo zabezpieczenie dawane przez RSA będzie wystarczające.

31. Omów połączenie RSA i DES. Dlaczego się je stosuje?

32. Omów algorytm ElGamala

Jest to algorytm asymetryczny bazujący na technikach wywodzących się z różnorodnych obserwacji matematycznych.

Własności:

- każdorazowo szyfrowanie wykorzystuje losowo wygenerowaną liczb,.
- klucz publiczny jest kluczem do szyfrowania, klucz prywatny jest kluczem deszyfrującym,
- kryptogramy są dwukrotnie dłuższe niż teksty jawne,
- metoda pozwala się rozwinąć bądź zastosować w innym kontekście,
- metoda wykorzystuje trudności obliczania tzw. dyskretnych logarytmów.

Algorytm ElGamala: komponentami są

- liczba pierwsza *p*, taka, że obliczania logarytmów modulo *p* jest praktycznie niewykonalne,
- generator α dla Z_p,
- liczby *t* i β, takie, że $t<p-1$ oraz $\beta=\alpha^t$.

Na klucz publiczny składają się *p, α, β.* Liczba t jest kluczem prywatnym.

Szyfrowanie: załóżmy, że zaszyfrowana ma być liczba $x<p$. Wykonywane są następujące kroki:

- wybieramy losowo (i nie ujawniając rezultatu) liczbę $k<p$,
- obliczamy $y_1=\alpha^k \bmod p$ oraz $y_2=x*\beta^k \bmod p$
- para liczb (y_1, y_2) tworzy szyfr dla *x*.

Deszyfrowanie: $y_2*(y_1^t)^{-1}=x*\beta^k*(\alpha^{kt})^{-1}=x*\alpha^{tk}*(\alpha^{tk})^{-1}=x \bmod p$. Ponieważ osoba deszyfrująca zna *t*, może więc wyznaczyć liczbę y2*(yt1)-1mod p a tym samym *x.*

33. Na czym polega teoretyczne i praktyczne bezpieczeństwo szyfrów

Bezpieczeństwo kryptosystemów można rozpatrywać z dwóch stron:

1. Bezpieczeństwo teoretyczne – bezwarunkowe, takich kryptosystemów jest niewiele, teoria nie zapewnia bezpieczeństwa. Jeśli istniałyby tak bezpieczne szyfry, to można stwierdzić, że nie ma sposobu, by je złamać, znaleźć klucz. Kryptoanalityk teoretycznie dysponuje nieograniczoną ilością czasu i sprzętu obliczeniowego. Jest w stanie przeanalizować wszystko w czasie zerowym.
2. Bezpieczeństwo praktyczne – uwarunkowane możliwościami obliczeniowymi współczesnej kryptoanalizy, daje dokładne wyniki szybkości działania kryptoanalizy. Dysponujemy konkretną mocą obliczeniową i czasem, możemy wtedy mówić o bezpieczeństwie obliczeniowym, algorytm jest bezpieczny jeśli dysponujemy takimi warunkami. Można w ten sposób również podać, które kryptosystemy są całkowicie bezpieczne na obecna chwilę.

Te dwa podejścia są całkowicie rozłączne.

34. Podaj definicję szyfru doskonałego

Kryptosystem jest to piątka (P, C, K, E, D), gdzie: P jest zbiorem możliwych jednostek tekstu jawnego, C jest zbiorem jednostek szyfrogramu, K to przestrzeń klucza, E jest szyfrogramem, D – deszyfrogramem. Znając te wartości możemy znaleźć ilość możliwych tekstów jawnych M, rozmiar przestrzeni klucza N oraz dla każdego tekstu jawnego prawdopodobieństwo wystąpienia danej litery w tekście jawnym $p(M_1)$. Podobnie możemy ustalić prawdopodobieństwo pojawienia

$\forall_{i,j} p(M_i \mid C_j) = p(M_i)$

się każdego z możliwych kluczy. Mając te dane w prosty sposób można zdefiniować kryptosystem doskonały:

Szyfr nazywamy doskonałym wtedy i tylko wtedy, gdy:

Dla wszystkich możliwych szyfrogramów i tekstów jawnych musi zachodzić: prawdopodobieństwo, że wystąpi dany tekst jawny Mi pod warunkiem, że wybraliśmy w szyfrogramie Cj musi być takie samo jak prawdopodobieństwo wystąpienia samego szyfrogramu, tekstu jawnego (tej litery w tekście jawnym), tzn., że na podstawie przechwyconego szyfrogramu nie jesteśmy w stanie powiedzieć jaki tekst jawny odpowiada takiemu szyfrogramowi lub jest mu prawdopodobny.

Cechy:

-nie ujawnia żadnej informacji o tekście jawnym;

-jest odporny (bezwarunkowo) na atak ze znanym szyfrogramem;

- spełnia $1 \geq n$ zależność (więcej kluczy niż wiadomości);

Te wymagania spełnia jedynie algorytm z kluczem jednokrotnym.

Jedynym znanym szyfrem doskonałym jest szyfr jednorazowy.

Szyfr doskonały jest odporny na atak ze znanym szyfrogramem niezależnie od mocy obliczeniowej posiadanej przez atakującego. W szyfrze doskonałym liczba wszystkich kluczy jest nie mniejsza (≥) od liczby wszystkich wiadomości jawnych. W wyidealizowanej sytuacji szyfr Cezara jest szyfrem doskonałym, jeśli każda litera jest szyfrowana jedną literą (losowo wybraną literą).

Szyfr Vernama jest szyfrem doskonałym.

35. Scharakteryzuj pojęcie odstępu jednostkowego

Odległość jednostkowa – odległość od jednoznacznego rozwiązania.

W języku naturalnym możliwymi tekstami jawnymi są ciągi liter. Łatwo zauważyć, że nie wszystkie takie ciągi są możliwe (mają sens). Takie ciągi są redundantne – nadmiarowe.

Jeżeli dla #M = n nie wszystkie ciągi liter są dozwolone, to znaczy nie wszystkie stanowią sensowne teksty jawne, to nazywamy je redundantnymi (np. ciągi liter języka naturalnego).

W każdym języku naturalnym występuje redundancja, czyli nadmierność.

Redundancję źródłową definiujemy jako:

D=H(C) − H(M), różnicę między entropią liter szyfrogramu i entropii tekstów jawnych, gdzie H(C) i H(M) są entropiami C i D dla pojedynczych liter.

Mając redundancję źródłową można podać definicje odległości jednostkowej – jest to iloraz entropii klucza i redundancji źródłowej.

Jeśli entropia szyfrogramu i tekstu jawnego jest równa 0, to odległość jednostkowa jest nieskończona.

Jeśli dla kryptosystemu odległość jednostkowa jest nieskończona to wtedy mówimy, że daje on idealną tajność (czyli redundancja klucza musi być duża, lub redundancja źródłowa jest bliska zeru).

Własności:

Odległość jednostkowa N:

- taka przybliżona długość szyfrogramu, że suma rzeczywistej informacji (entropii) w odpowiednim tekście jawnym i entropii klucza jest równa liczbie bitów szyfrogramu;
- liczba bitów szyfrogramu wystarczająca do jednoznacznego wyznaczenia klucza, przy założeniu posiadania nieograniczonej mocy obliczeniowej i nieograniczonego czasu;
- nie jest miara tego jak długi szyfrogram jest potrzebny do kryptoanalizy, lecz jak długi szyfrogram jest konieczny;

36. Podaj definicję szyfru idealnego

Kryptosystem idealny nie musi być doskonały, ale kryptosystem doskonały musi być idealny. Jeżeli kryptosystem jest idealny, to nawet po udanej kryptoanalizie pozostaje wątpliwość, czy uzyskany tekst jawny jest prawdziwy.

37. Omów techniki stosowane w celu zmniejszenia nadmiarowości w kryptosytemach

Mieszanie i rozpraszanie – służą do zmniejszenia nadmierności.

Mieszanie: cel: własności szyfrogramu nie takie same jak własności tekstu jawnego.

Metody: podstawianie: na blokach kilkuznakowych lub kilkubitowych, szyfry polialfabetyczne, skrzynki podstawieniowe

Rozpraszanie: cel: nadmiarowość tekstu jawnego rozprzestrzenić po całym szyfrogramie.

Metody: permutacja: transpozycja uzależnienie jednego znaku szyfrogramu od jak największej ilości bitów tekstu jawnego; szyfry strumieniowe – mieszanie; szyfry blokowe – mieszanie i rozpraszanie

38. Jakie warunki musi spełniać jednokierunkowa funkcji skrótu?

Argumentem funkcji H(M) jest wiadomość M o dowolnej długości. Wartością funkcji jest liczba h o ustalonej długości. Jednokierunkowe funkcje skrótu mają następujące własności:

Mając dane M, łatwo jest obliczyć h

Mając dane h, trudno jest obliczyć M
Mając dane M, trudno jest znaleźć inną wiadomość M' taką, że H(M)=H(M')
W większości implementacji problem uważa się za trudny, jeżeli wymaga więcej niż 264 operacji. SHA wytwarza skrót o długości 160 bitów.

39. Podaj przykłady funkcji skrótu wraz z długościami wytwarzanych skrótów

Jednokierunkowe funkcje skrótu:

- funkcja Snerfu –128 lub 256 bitów
- funkcja N-Hash – 128 bitów
- funkcja MD2, MD4, MD5 – 128 bitów
- bezpieczny algorytm skrótu – 160 bitów
- funkcja RIPE-MD (odmiana MD4)
- funkcja HAVAL – 128,160,192, 224 bity
-

40. Omów atak wykorzystujący paradoks urodzin. Jakie są bezpieczne aktualnie długości skrótów?

41. Co nazywamy protokołem kryptograficznym? Po co się je stosuje

Kryptosystemy mają zastosowanie w protokołach kryptograficznych, które umożliwiają pewne usługi w wymianie informacji.

Protokół – szereg kroków obejmujących dwie lub więcej stron podejmowanych w celu realizacji zadań.

Własności:

1. Każdy użytkownik protokołu musi znać wszystkie kroki protokołu
2. Użytkownik musi zgodzić się na jego stosowanie.
3. Protokół musi być nie mylący, każdy krok powinien być dobrze zdefiniowany i nie może wystąpić jakakolwiek szansa na nieporozumienie.
4. Protokół musi być kompletny: dla każdej możliwej sytuacji musi być podany odpowiedni sposób postępowania.

Cel stosowania protokołów:

1. Wymiana jakiejś liczby losowej w celu generowania klucza;
2. Podzielenie się tajemnicą tak, by nikt inny nie miał do niej dostępu. Wymiana informacji w sposób bezpieczny.

Protokół kryptograficzny – protokół wykorzystujący kryptografię.

Strony uczestniczące w protokole mogą być bezwarunkowo ufającymi sobie osobami, lub w ogóle nie ufać sobie, mogą być przeciwnikami. Chodzi o to by obie strony osiągnęły swój cel.

42. Wymień typy protokołów kryptograficznych i rodzaje ataków na te protokoły

Arbitrażowe – uczestniczy w nich trzecia strona – arbiter. Jest to osoba nie zainteresowana, obdarzona zaufaniem. Dzięki tej osobie uczestnicy protokołu są pewni poprawnego przebiegu protokołu. Warunkiem poprawnego zakończenia protokołu jest uczestnictwo arbitra.

Rozjemcze – wpierw wykonujemy protokół nie arbitrażowy z udziałem dwóch stron. Protokół rozjemczy wykonywany jest jedynie w wyjątkowych okolicznościach, wtedy gdy toczy się spór między stronami protokołu. Arbiter biorący udział w protokole rozjemczym nazywany jest sędzią – nie jest zatrudniony w każdym protokole. Arbiter rozstrzyga, która ze stron złamała protokół lub gdzie wystąpiły błędy.

Samowystarczające – protokół taki jest bardzo trudny do zrealizowania. Jest on tak sformułowany, że obie strony wymiany nie mają możliwości popełnienia błędu.

Łamanie protokołów

Nawet jeśli protokół używa dobrego kryptosystemu, kroki w nim przewidziane mogą być źle skonstruowane i łatwo go złamać. Do próby łamania protokołu może się również przyczynić kryptosystem w który protokół jest zapisany.

Atak bierny – przeciwnik przesłuchuje wszystkie informacje w trakcie działania protokołu i na tej podstawie próbuje zdobyć informacje o którejś ze stron lub przesyłane informacje.

Atak czynny – przeciwnik stara się odnieść pewne korzyści poprzez ingerencję w protokół, np. wkłada do protokołu nowe komunikaty, podszywa się pod którąś ze stron, wysyła jeszcze raz to, co przechwycił, próbuje usunąć któryś z komunikatów.

Łamanie czynne jest bardziej groźne, szczególnie dotyczy to protokołów. Oponent ma dużo więcej możliwości – może nim być któraś ze stron. Takiego oponenta nazywamy oszustem.

Oszust bierny – jest to jedna ze stron, która dąży do złamania protokołu, dla własnej korzyści, przeważnie działa zgodnie z protokołem, ale próbuje przechwycić informacje o drugim uczestniku, stara się złamać klucz, zdobyć dane.

Oszuści aktywni – nie wykonują prawidłowo protokołu.

43. Omów usługi związane z ochroną informacji

Poufność – zabezpieczenie wymiany informacji między stronami przed ujawnieniem.

Integralność danych – przeciwnik nie jest w stanie modyfikować informacji, np. dołożyć, usunąć, zmienić kolejność komunikatu. Usługa ta ma zapewniać ochronę przed błędami transmisji.

Integralność danych – przeciwnik nie jest w stanie modyfikować informacji, np. dołożyć, usunąć, zmienić kolejność komunikatu. Usługa ta ma zapewniać ochronę przed błędami transmisji.

Niezaprzeczalność – żadna z osób nie wyparła się po transmisji, że komunikacja niedoszła do skutku, np. przy zawieraniu umów (tu często stasuje się podpis cyfrowy).

44. Podaj usługi realizowane przez podpis cyfrowy i schematy przykładów podpisu

Podpis cyfrowy jest bardzo istotnym narzędziem. Jest to protokół dzięki któremu możemy realizować wiele usług: uwierzytelnienie, integralność danych i niezaprzeczalność.

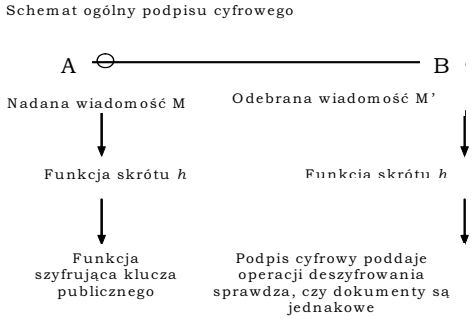
Podstawowym zastosowaniem podpisu cyfrowego jest certyfikacja kluczy publicznych.

Podpis zwykły (ręczny)	Podpis cyfrowy
Cechy wspólne	
Możliwość przypisania tylko jednej osobie	
Podpis może formalnie wykonać tylko jedna osoba	
Autor nie wyprze się swojego podpisu	
Różnice	
Jest składany łącznie z dokumentem	**Składany niezależnie od dokumentu**
Taki sam dla wszystkich dokumentów	Może być dla każdego dokumentu inny
Składany na ostatniej stronie dokumentu	Może być na całym dokumencie, dzięki czemu można wykryć zmiany w dokumencie

Schematy podpisów cyfrowych.

1. Wykorzystanie algorytmu klucza publicznego RSA. Nadawca poddaje dokument funkcji skrótu. Otrzymany skrót poddaje operacji potęgowania – wykorzystując klucz prywatny – mod2, zgodnie z RSA . powstały skrót dołącza do wysyłanej wiadomości. Odbiorca wylicza ta sama funkcje skrótu – posługuje się kluczem publicznym nadawcy, sprawdza czy otrzymany skrót odpowiada otrzymanemu podpisowi cyfrowemu.

2. Wykorzystanie algorytmu ElGamala. Nadawca poddaje dokument funkcji



skrótu – wykorzystuje klucze *pgx*, zgodnie ze schematem ElGamala. Wybiera po funkcji skrótu p_k i wylicz wartości r_j, które stanowią podpis cyfrowy dla wiadomości. Tak skonstruowaną instrukcje przesyła odbiorcy. Odbiorca używając klucza publicznego nadawcy, sprawdza ważność podpisu cyfrowego.

3. Schemat DSA – schemat podpisu cyfrowego. Do dokumentu nadawca wykonuje funkcję skrótu czwórką *pqgx*, wykonuje obliczenia i wyznacza parametry. Stanowią one podpis cyfrowy. Odbiorca zna klucz publiczny nadawcy, i posługuje się czwórką *pqgy* by wykonać obliczenia. Na podstawie otrzymanych parametrów może sprawdzić podpis.

45. Przedstaw metody realizacji integralności danych

Integralność danych – podpis cyfrowy realizuje nam również integralność danych, którą można uzyskać poprzez dołączenie pewnych wiadomości kontrolnych, nadmiarowych. Najczęściej wykorzystuje się: sumy kontrolne, bity parzystości, funkcje jednokierunkowe dzięki nim modyfikacje lub błędy w przekazanych dokumentach:

1. Używając funkcji EMAC (do obliczenia skrótu potrzebny jest tajny klucz). Dla wiadomości generujemy taką funkcje skrótu. Znając klucz dołączamy do wygenerowanej przez nas wiadomości. Wiadomość przesyłamy kanałem jawnym.
2. Funkcja skrótu + szyfrowanie – najpierw wykonujemy skrót pewnym algorytmem, na końcu skrót ten poddajemy szyfrowaniu z jakimś tajnym kluczem.
3. Funkcja skrótu + bezpieczny kanał – wysyłamy osobno wiadomość i osobno bezpiecznym kanałem skrót.

46. Przedstaw metody realizacji identyfikacji (uwierzytelnienia)

Identyfikacja i uwierzytelnienie mogą być traktowane jako jedna usługa, np. dokonujemy identyfikacji informacji a następnie jej uwierzytelnienia.

Identyfikacja – dopuszczamy strony do wymiany informacji.

Cel: potwierdzenie tożsamości stron wymiany informacji. Chodzi o to by tak dokonać tej identyfikacji by strona B nie mogła podszyć się pod A. W tym celu można zastosować hasło, szyfrowanie kilku haseł i funkcję skrótu.

Atak: Słownikowy –przeciwnik bada czy podane słowa, ciągi znaków poddane funkcji skrótu odpowiadają skrótowi jaki przechwycił. Konstruuje słownik i bada czy któreś z haseł odpowiada tym zawartym w pliku haseł.

1. Hasło jednokrotne – algorytm haseł jednokrotnych Lamporte'a, który wykorzystuje funkcje jednokierunkowe – każde hasło generujemy na podstawie poprzedniego. Oponent nie jest w stanie zgadnąć hasła.
2. Identyfikacja wyzwanie (challenge) – osoba daje wyzwanie – hasło (przeważnie jest to liczba) - i oczekuje na nie konkretnej odpowiedzi.
3. Algorytm identyfikacji Schnorra

47. Omów rodzaje kart inteligentnych

Inteligentne karty są to plastikowe karty z układem logicznym, posiadają pamięć.

Karty przechowujące dane – mają zapisane w sobie dane np. o użytkowniku. Karty te mają większą niezawodność. Z uwagi na małą pojemność nie mogą być stosowane jako karty bankowe.

Karty telefoniczne – mają układ logiczny, który umożliwia jedynie zmniejszenie wartości karty.

Karty identyfikacyjne – stosowane w momencie dostępu do danych – karta wykorzystuje protokół identyfikacji, wyzwania (challenge) – karta komunikuje się z czytnikiem, odpowiada na zadawane pytania, jeśli odpowiedzi okażą się prawdziwe właściciel ma prawo wejścia do danego obiektu (dokumentu). Takie karty mają pewien klucz, np. PIN, numer którego wczytanie pozwala na komunikację.

Elektroniczna portmonetka – możemy ją wielokrotnie ładować, używać w każdym czasie. Jest ona odpowiednio zabezpieczona, by ktoś, kto np. nie ma funduszy, nie naładował takiej karty i z niej nie korzystał. Karta taka operuje na niewielkich kwotach.

48. Jakie usługi realizuje i jakie algorytmy wykonuje program PGP?

PGP jest programem do wysyłania wiadomości. Realizuje następujące usługi: poufność, uwierzytelnienie źródła pochodzenia wiadomości, spójność, integralność danych.

PGP używa następujących systemów kryptograficznych: IDEA w trybie CBC i RSA – do generowania i dystrybucji kluczy.

Wysyłanie wiadomości następuje w 4 etapach:

1. skrót MD5 wiadomości; szyfrowanie skrótu za pomocą RSA, kluczem prywatnym nadawcy;
2. kompresja wiadomości wraz z podpisem cyfrowym programem np. ZIP 2.0
3. uzupełnienie do bloków 8-bitowych; wytworzenie losowego klucza i wektora IV; szyfrowanie wiadomości algorytmem IDEA w trybie CBC; szyfrowanie klucza i IV RSA kluczem jawnym odbiorcy (jeśli wysyłamy wiadomość do wielu odbiorców to szyfrujemy klucz IDEI wszystkimi kluczami jawnymi odbiorców)
4. kodowanie ciągu binarnego na kod ASCII algorytmem Radix-64; dołączenie kodu detekcji błędów transmisji.

PGP umożliwia segmentację wiadomości na mniejsze części i po otrzymaniu ich połączenie w jedną całość.